

# Motion Planning in Non-Gaussian Belief Spaces (M3P): The Case of a Kidnapped Robot

Saurav Agarwal

Amirhossein Tamjidi

Suman Chakravorty

**Abstract**—Planning under uncertainty is a key requirement for physical systems due to the noisy nature of actuators and sensors. Using a belief space approach, planning solutions tend to generate actions that result in information seeking behavior which reduce state uncertainty. While recent work has dealt with planning for Gaussian beliefs, for many cases, a multi-modal belief is a more accurate representation of the underlying belief. This is particularly true in environments with information symmetry that cause uncertain data associations which naturally lead to a multi-modal hypothesis on the state. Thus, a planner cannot simply base actions on the most-likely state. We propose an algorithm that uses a Receding Horizon Planning approach to plan actions that sequentially disambiguate the multi-modal belief to a uni-modal Gaussian and achieve tight localization on the true state, called a Multi-Modal Motion Planner (M3P). By combining a Gaussian sampling-based belief space planner with M3P, and introducing a switching behavior in the planner and belief representation, we present a holistic end-to-end solution for the belief space planning problem. Simulation results for a 2D ground robot navigation problem are presented that demonstrate our method's performance.

## I. INTRODUCTION

Motion planning for robotics involves dealing with the uncertain nature of physical systems i.e. noisy actuators and sensors as well as changes in the environment in which the robot operates. The motion or actuator uncertainty makes it difficult to execute precise actions and sensing uncertainty makes it impossible to determine the exact state of the robot. Further, changes in the environment can reduce the effectiveness of plans computed offline. Thus, unless a plan can be updated on the fly to account for new constraints, the plan might fail. A significant amount of research has gone into developing probabilistic methods to achieve robust performance for practical systems. In the probabilistic approach, the aim is to develop methods that maximize the probability of achieving a desired state. State of the art methods rely on a probability distribution over the system's state (called the belief) and develop solutions in the belief space that enable us to drive the system belief from an initial to a desired belief. In the case of a mobile robot, we may wish to drive the system from a start location to a goal location, or in the case of a manipulator, manipulate an object from an initial to some final state. In general, planning for systems under uncertainty belongs to the class of Partially-Observable

Markov Decision Process (POMDP) problems which are known to be computationally intractable.

Recent work, in particular, sampling based methods, have shown varying degrees of success in solving the POMDP problem. In [14], the authors construct a belief space variant of a Probabilistic RoadMap (PRM) [7] to generate plans that result in information gathering actions which minimize state uncertainty at the goal. The authors use linearized process and measurement models, and a Gaussian belief parametrization. In [4], a graph is constructed in belief space and pruned successively, ultimately resulting in a tree in belief space which guarantees convergence to the optimal solution in the limit of infinite samples. These methods along with others [5], [16], [8] provide solutions that are dependent on the initial belief. This makes them computationally inefficient for real-time replanning. Feedback-based Information RoadMap (FIRM) [3] also builds a belief space variant of a PRM, but this work differs from the previous ones in that it enables re-use of the offline computations to compute new policies online. FIRM introduces belief stabilizers at the graph nodes which act as funnels and lead to edge independence in the FIRM graph. We note that the different instantiations of this method also assume a Gaussian belief representation. This method is shown to work well in practice in [2] on a mobile robot operating indoors.

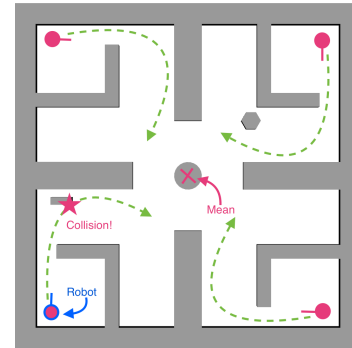


Fig. 1: A kidnapped robot scenario in a world with 4 identical rooms. The actual robot is depicted by the blue outlined disk, whereas the hypothesis are depicted by red disks. All hypothesis are equally likely. The dashed green arrows show a possible control action based on the hypothesis in the top-right room that can result in collision for the actual robot.

### A. Issues with Gaussian Belief Representation

In [2] situations such as kidnapping (lost-robot), were dealt with by expanding the error covariance such that the Bayesian estimator (in this case a Kalman filter) is able to account for the unexpected data associations/observations or large innovations. This works well as long as there are no uncertain data associations. The robot was able to enter into an Information Gathering Mode and make small cautious movements to gain enough information until the error covariance converged to a small threshold. Certain situations can arise where data association between what is observed and the robot's a-priori map leads to a multi-modal hypothesis. An example is the **data association problem** for a kidnapped robot operating in a symmetric environment. Imagine a mobile robot equipped with a laser scanner operating in a world where there are identical rooms as shown in Fig. 1. To the laser, each and every room appears identical. Thus, if the robot is switched off and placed randomly in one of the rooms, on waking up it has no way of knowing exactly where it is. In a sense, based on the observations, the robot would think it could be in either of the four rooms at that instant since the laser readings would look alike. We can extend the previous statement to say that when sensory information leads to uncertain data associations, the robot may believe itself to be in one of multiple states at the same time. Such a situation implies that the pdf of the robot's belief cannot be represented by a uni-modal Gaussian.

In [12], the authors investigate the grasping problem with a multi-modal hypothesis of the gripper's state. Their method picks the most-likely hypothesis and a fixed number of samples from the belief distribution, then using a Receding Horizon Control approach, belief space trajectories are found using direct transcription that maximize the observation gap between the most-likely hypothesis and the drawn samples. The authors use this approach to prove/disprove the most likely hypothesis. In [13], the correctness and complexity of the algorithm presented in [12] is analyzed and proven to eventually converge to the goal region in belief space with as low as two samples. In [11], the authors build upon [12] wherein they transpose the non-convex trajectory planning problem in belief space to a convex problem that minimizes the average log-weight of the belief-modes and a quadratic action cost.

Fundamentally, our work and the work in [11, 12] aim to achieve the same goal of proving/disproving hypothesis of the robot's state. At the very basic level, these methods choose actions that lead to information gathering behavior such that different modes of the robot's belief are expected to observe different information, thus disambiguating the multi-modal hypothesis. In our method, we develop an information graph node based receding horizon planner that sequentially disambiguates the different hypothesis encoded in the multi-modal belief state. Further, the technique developed is guaranteed to drive the belief state into a unimodal belief state from

where a Gaussian planner such as FIRM can take over to complete the robot's mission. We believe that our method is computationally more efficient than the optimization based technique in [11, 12]. It is also able to deal with the kidnapped robot scenario which may not be possible to address using the trajectory optimization based technique in [11, 12] due to the difficulty of generating an initial feasible plan for the widely separated modes in the presence of obstacles. We further believe that our planner is applicable to any situation wherein a multi-modal belief may arise in the robot's state due to uncertain data associations. We also show how the multi modal planner can be seamlessly combined with a unimodal belief space planner such as FIRM to facilitate belief space planning for robotic tasks where Gaussian/ unimodal belief representations may not be valid during the entirety of the task.

**Contributions:** The key contributions of our work are as follows:

- 1) We represent the belief with a Gaussian Mixture Model (GMM) rather than particles.
- 2) Instead of basing actions on the most-likely hypothesis, we create candidate actions based on each mode and evaluate the best one.
- 3) We use a sampling based planner i.e. RRT\* [6] to plan candidate trajectories. (One can also simply use RRTs [10] but due to insignificant overhead in using RRT\* over RRT we prefer RRT\* as it gives us the benefit of optimality)
- 4) We introduce a switching behavior in the belief representation during the online-phase from Gaussian to non-Gaussian, and back, as required. Our argument is that most of the times, the belief is well represented by a Gaussian, wherever this is not the case, we switch to a GMM and our algorithm creates plans that converge back to a uni-modal Gaussian.
- 5) We present simulation results for a 2D navigation problem in which a robot is kidnapped.

## II. PROBLEM STATEMENT

Let  $x_k$ ,  $u_k$ , and  $z_k$  represent the system state, control input, and observation at time step  $k$  respectively. Let  $\mathbb{X}$ ,  $\mathbb{U}$ , and  $\mathbb{Z}$  denote the state, control, and observation spaces respectively. It should be noted that in our work, the state  $x_k$  refers to the state of the mobile robot i.e. we do not model the environment and obstacles in it as part of the state. The sequence of observations and actions are represented as  $z_{i:j} = \{z_i, z_{i+1}, \dots, z_j\}$  and  $u_{i:j} = \{u_i, u_{i+1}, \dots, u_j\}$  respectively. The non-linear state evolution model  $f$  and measurement model  $h$  are denoted as  $x_{k+1} = f(x_k, u_k, w_k)$  and  $z_k = h(x_k, v_k)$ , where  $w_k \sim \mathcal{N}(0, Q_k)$  and  $v_k \sim \mathcal{N}(0, R_k)$  are zero-mean Gaussian process and measurement noise, respectively.

We assume that the robot is initially tasked to go from a start location to a goal location. For this purpose, we use a Gaussian Belief Space Planner such as FIRM. However, during

the course of the task, say at time  $k$ , the robot is kidnapped, or gets lost, leading to uncertain data associations for its sensor readings. In such a case, the belief  $b_k$  can be represented by a GMM at time  $k$  as a weighted linear summation over Gaussian densities. Let  $w_{i,k}$ ,  $\mu_{i,k}$  and  $\Sigma_{i,k}$  be the importance weight, mean vector and covariance matrix associated to the  $i^{th}$  Gaussian  $m_i$  respectively at time  $k$ , then

$$b_k = \sum_{i=1}^N w_{i,k} m_{i,k}, \quad m_{i,k} \sim \mathcal{N}(\mu_{i,k}, \Sigma_{i,k}). \quad (1)$$

*Our goal is to construct a belief space planner  $\mu(b_k)$  such that under the belief space planner, given any initial multi-modal belief  $b_0$ , the belief state process evolves such that  $b_T = m_T$ , where  $m_T = \mathcal{N}(\mu_T, \Sigma_T)$  for some finite time  $T$ .*

In other words, our goal is to construct a belief space planner such that it is guaranteed to drive the initial multi-modal belief into a unimodal belief in finite time. The basic idea is that once such a unimodal belief is achieved, a Gaussian belief space planner such as FIRM can take over again and guide the robot towards its mission goal which was interrupted due to the kidnapping.

Note that we do not require optimality from our planner, only that it stabilize the belief state process to a unimodal belief in a guaranteed fashion. Further, albeit the scenario used to motivate the problem is the kidnapped robot situation, the method proposed is general, and can be extended to any planning situation where a multi-modal belief arises in the robot state due to uncertain data associations.

### III. METHODOLOGY

One cannot simply base actions on the mean, as in a multi-modal scenario, it would not make sense in the physical world. Again, taking the example shown in Fig. 1, the mean of all the modes may lie within some obstacle which obviously is not a good choice to base actions on. Our algorithm creates candidate plans for each belief mode that guide it to a state called the target state such that the targets of all the modes have minimal information overlap. For each policy, we simulate the expected information gain which we define as the reduction in the discrete number of modes and choose the gain maximizing policy.

#### A. Belief Propagation Using Gaussian Mixture Model

We adopted the Gaussian-Mixture-Model (GMM) for belief representation because it provides a seamless transition from multiple to single hypothesis scenarios and vice versa during localization and planning. We found in our experiments that the particle filter alternative [9] which is generally used for global localization in a kidnapped situation is slower in convergence and less robust compared to the approach presented here. Our method also incurs less computational cost compared to PF-Localization. Once the lost state is detected, we need to

generate a rich set of samples from which we can converge to the most likely modes. We uniformly sample the configuration space and set these samples as the means  $\mu_{i,k}$  of the modes of the Gaussian and assign equal covariance to each mode. Algorithm 1 shows how we generate the initial multi-modal belief. Note that  $n_k$ , the number of Gaussian-modes at time  $k$ , can vary depending on the observational sequences as new hypotheses are added and some die out. Further, we keep the weights normalized such that  $\sum_{i=1}^N w_{i,k} = 1$ . As the robot moves and gets observations, the importance weights  $w_{i,k}$ 's are updated based on the measurement likelihood function as shown in Eq. 2.

$$w_{i,k+1} = \frac{w_{i,k} e^{-0.5 D_{i,k}^2}}{\sum_{i=1}^N w_{i,k} e^{-0.5 D_{i,k}^2}} \quad (2)$$

where  $D_{i,k}$  is the Mahalanobis distance between the sensor observation and most-likely observation for mode  $m_i$  such that

$$D_{i,k}^2 = (h(x_k, \nu_k) - h(\mu_i, 0))^T R_k^{-1} (h(x_k, \nu_k) - h(\mu_i, 0)). \quad (3)$$

Depending on the state of the robot, individual hypotheses and data association results, we might have several cases.

- 1) The number of real landmarks observed  $n_z$  could be equal, less or more than the number of predicted observations  $n_{h_i}$  where  $h_i$  is the predicted observation vector for the  $i^{th}$  mode.
- 2) The number of associated real and predicted observations  $n_{z \cap h}$  could be equal or less than  $n_z$ ,  $n_h$  or both.

Thus, to update the weight of each hypothesis, we factor in the above information. First we estimate the Mahalanobis distance  $D_{i,k}$  between the predicted and observed landmarks that are matched by the data association module. Then we multiply this weight by a factor  $\gamma$ , which models the effect of duration for which the robot and a hypothesis predict to see different information. The entire weight update algorithm is described in algorithm 2. After each weight update step, we remove modes with negligible contribution to the belief i.e. when  $w_{i,k+1} < \delta_w$  where  $\delta_w$  is a user defined parameter for minimum weight threshold (1% of the total weight in our simulations).

#### B. Multi-Modal Motion Planner (M3P)

The task of the non-Gaussian planner is to generate a sequence of control actions such that the belief state converges to a uni-modal Gaussian. We call our planner **M3P** i.e. Multi-Modal Motion Planner. Algorithm 3 describes the planner's key steps and Fig. 2 describes the basic functionality of the planner in a sample scenario.

The key steps in the M3P planner are steps 3 and 4 in Algorithm 3 that generate the candidate policies for the different modes, and then choose one of the policies based on the expected information gain. In the following section, we give an uniqueness graph (node) based instantiation of

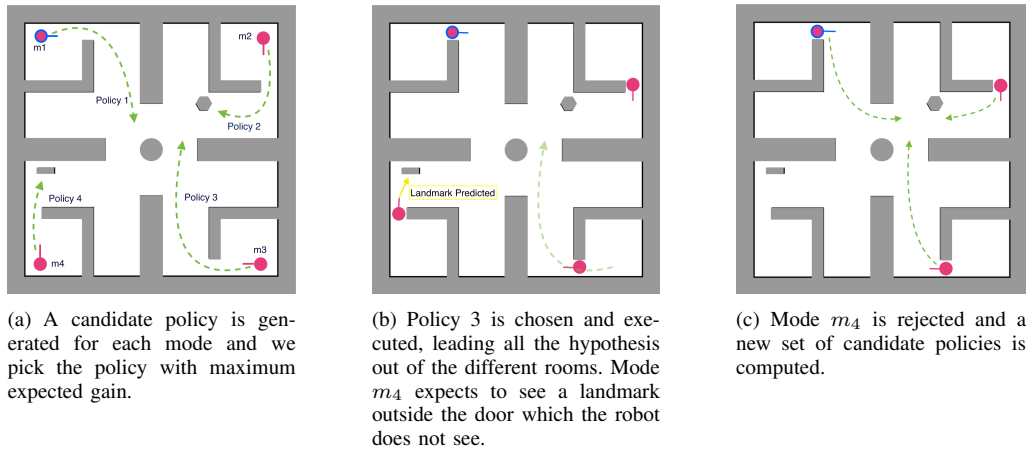


Fig. 2: Key steps in the Receding Horizon Control methodology for a multi-modal belief.

---

**Algorithm 1: Generating Multi-Modal Belief State**

---

```

1 Input:  $\Sigma_0$  (Initial Covariance)
2  $\{\mu\}$  = Sample valid states uniformly;
3 for  $\mu_i \in \{\mu\}$  do
4    $m_i$  = Compose belief mode with mean  $\mu_i$  and
   covariance  $\Sigma_0$  ;
5   Add mode  $m_i$  to belief  $b$  ;
6 Assign uniform weight to all modes ;
7 while not converged to fixed number of modes do
8   Update mode weights in belief  $b$ ;
9   Remove modes with weights lower than threshold  $\delta_w$ ;
10 return  $b$ ;

```

---



---

**Algorithm 2: GMM Weight Update**

---

```

1 Input:  $w_{i,k}, \mu_{i,k+1}$ 
2 Output:  $w_{i,k+1}$ ;
3  $z_{k+1}$  = Get sensor observations ;
4  $n_z$  = Number of landmarks observed in  $z_{k+1}$ ;
5  $n_{z \cap h}$  = Data association between  $h(\mu_{i,k+1}, 0)$  and  $z_{k+1}$  ;
6  $w'_{i,k+1}$  = Update and normalize weight according to
likelihood function ;
7 if  $n_h \neq n_z$  ||  $n_h \neq n_{z \cap h}$  then
8    $\alpha = \max(1 + n_z - n_{z \cap h}, 1 + n_h - n_{z \cap h})$  ;
9    $\beta = \beta + \delta t$  ;
10   $\gamma = e^{-0.0001\alpha\beta}$  ;
11 else
12   $\beta = \beta - 1.0$  ;
13  if  $\beta < 0$  then
14     $\beta = 0$ ;
15  $w_{i,k+1} = w'_{i,k+1}\gamma$  ;
16 return  $w_{i,k+1}$ ;

```

---



---

**Algorithm 3: M3P: Multi-Modal Motion Planner**

---

```

1 Input:  $b$  (Belief)
2 while  $b \neq \mathcal{N}(\mu, \Sigma)$  do
3    $\Pi$  = Generate candidate policy for each belief mode ;
4    $\pi$  = Pick policy from  $\Pi$  with maximum expected
information gain ;
5   forall the  $u \in \pi$  do
6      $b$  = Apply action  $u$  and update belief;
7     if Change in number of modes || Expect a belief
mode to violate constraints then
8       break;
9 return  $b$ ;

```

---

scenarios.

*C. Node based Multi-Modal Motion Planner (NBM3P)*

Here we give a particular instantiation of the two key steps required to enable the M3P planner; 1) generating the candidate policies and, 2) picking the optimal policy.

1) *Generating a Set of Candidate Policies:* In a multi-modal scenario no-one action based on a belief mode can be said to be the best action. The best action for one mode may or not be good for the other modes. Thus, in a multi-modal scenario, we claim that the best action to take is one that guides the robot without collision through a path that results in information gain such that the belief converges to a uni-modal Gaussian pdf. We reduce the infinite set of possible control actions to a discrete number of policies by picking a target state for each mode to go to and generate control actions that guide each mode to its corresponding target.

*Picking the target state for a mode:* We introduce the concept of a uniqueness graph  $G_u$  (computed offline) that allows us to pick target states for each belief mode. The uniqueness graph  $G_u$  is constructed using information from

the planner which allows us to recover from kidnapped robot

the FIRM graph. Each FIRM node is added as a node in  $G_u$  (only  $x, y, \theta$ ). Once a node is added, we calculate what information the state represented by that node can observe in the environment. Using this information we add an edge  $E_{\alpha\beta}$  (undirected) between two nodes  $v_\alpha$  and  $v_\beta$  if both nodes see similar information. In our landmark based observation model, each landmark has an ID. Thus if both nodes observe the same ID or IDs then an edge is added and the weight of the edge is equal to the number of similar landmarks observed. The edge weight gives an indication of the similarity (or conversely uniqueness) in the information observed. If the weight of an edge is higher, it means the states represented by the vertices of that edge are more likely to observe similar information. To find the target for a belief mode  $m_i$ , we first choose the set of nodes  $N_i$  which belong to the neighborhood of  $\mu_i$  (which is the mean of mode  $m_i$ ). Then, we find the node  $v_i^t \in N_i$  which observes information that is least similar in appearance to that observed by nodes in the neighborhood  $N_j$  of mode  $m_j$  where  $j \neq i$ . We are trying to solve the optimization problem,

$$(v_1^t, v_2^t \dots v_n^t) = \arg \min_{v_1, v_2, \dots, v_n} (h(v_1, 0) \cap h(v_2, 0) \cap \dots \cap h(v_n, 0)). \quad (4)$$

To do this, first we calculate the total weight of the outgoing edges from every node  $v \in N_i$  to nodes in all other neighborhoods  $N_j$  where  $j \neq i$ . The node which has the smallest outgoing edge weight, is the target candidate. Algorithm 4 describes in detail the steps involved.

---

**Algorithm 4:** Finding the target for a mode in a multi-modal belief

---

```

1 Input:  $m_{i,k}$ ,  $G_u$  ; // belief mode, uniqueness graph
2 Output:  $v_i^t$ ; // target node
3 forall the  $m_{l,k} \in M_k$  do
4    $N_l = \text{Find nodes in } G_u \text{ within neighborhood of}$ 
    $\text{radius } R \text{ centered at } \mu_{l,k}$  ;
5  $\text{minWeight} = \text{Arbitrarily large value}$ ;
6  $v_i^t = -1$  ;
7 forall the  $v \in \text{Neighborhood } N_i \text{ of mode } m_{i,k}$  do
8   for  $N_j \in \text{set of all neighborhoods of the belief modes}$ 
   and } j \neq i do
9      $w = 0$  ;
10    forall the  $e \in \text{Edges connected to } v$  do
11      forall the  $p \in N_j$  do
12        if  $p \text{ is a target of edge } e$  then
13           $w += \text{edgeWeight}(e)$ ;
14    if  $w \leq \text{minWeight}$  then
15       $\text{minWeight} = w$ ;
16       $v_i^t = v$  ;
17 return  $v_i^t$ ;

```

---

*Connecting a mode to its target:* Once we have picked the targets corresponding to each mode, we need to find the control action that can take the mode from its current state to the target state. We generate the open loop control sequence that takes each mode to its target using the RRT\* planner. RRT\* is chosen because it is computationally cheap and can incorporate the system's kinodynamical constraints.

2) *Picking the Optimal Policy:* Once we have generated the set of candidate policies. We need to evaluate the expected information gain from each policy and pick the optimal policy that maximizes this information gain. We model this information gain as the discrete change in the number of modes. This implies that instead of doing Monte Carlo simulations with noisy observations and motion to generate all possible belief trajectories, we can simulate the most-likely belief trajectory. This helps reduce the computational burden significantly. We know that one policy may or may not be good for all the modes i.e. a policy based on one mode may lead to collision for the other modes. Therefore, we need a way of penalizing a candidate policy if it results in collision. We introduce a penalty  $c_{fail}/k$  where  $c_{fail}$  is a fixed value ( $10^6$ ) and  $k$  is the step during execution at which the collision takes place. Thus, policies which result in a collision much further down are penalized less compared to policies that result in immediate collision. The steps to calculate the expected information gain for a candidate policy are:

- 1) Pick a candidate policy  $\pi_i$  corresponding to mode  $\mu_i$
- 2) Assume that the robot is actually at one of the modes.
- 3) Execute the policy and propagate all the modes and measure the information gain (the reduction in the number of modes).
- 4) Repeat steps 2-3 for each mode and sum the weighted information gain weighted by the mode weights.

We repeat the above steps for all policies in the set  $\Pi$  and pick the policy with maximum weighted information gain.

Algorithm 5 shows how we generate the optimal policy  $\pi$ . Two key functions to note here are:

- 1)  $\text{findTargetState}(\bar{b}_k^i)$  : Finds a FIRM node in the neighborhood of mode  $\bar{b}_k^i$  at time  $k$  such that by going to this node, the robot can gain the maximum possible information. Algorithm 4 shows how this function works.
- 2)  $\text{calculateInformationGain}(\pi_i, \bar{b}_k^i)$  : Given a control sequence  $\pi_i$  and mode  $\bar{b}_k^i$ , it calculates the expected information gain. Algorithm 6 shows how the expected information gain is calculated for the candidate policies.

#### D. Analysis

In this section, we show that the basic receding horizon planner M3P will guarantee that an initial multi-modal belief is driven into a unimodal belief in finite time. First, we make the following assumptions:

*Assumption 1:* Given a multi-modal belief  $b_k = \sum_i w_{i,k} m_{i,k}$ , for every mode  $m_{i,k}$ , there exists a disambiguating planner  $\mu_i(\cdot)$  in the sense that if the robot was really in mode  $i$ , the planner's actions would confirm that the robot was at mode  $i$ .

---

**Algorithm 5:** Calculating the Optimal Policy  $k$ 

---

```
1 forall the  $m_{i,k} \in M_k$  do
2    $v_{i,k}^t = \text{Find target state for mode } (m_{i,k})$  ;
3    $\pi = \text{Find open loop controls from } \mu_{i,k} \text{ to } v_{i,k}^t \text{ using}$ 
    $\text{RRT}^*$  ;
4    $\Pi.\text{append}(\pi)$ ;
5 forall the  $\pi_j \in \Pi$  do
6   forall the  $m_{i,k} \in M_k$  do
7      $\Delta I_{ij} = \text{Calculate expected information gain for}$ 
        $\text{policy } \pi_j \text{ assuming robot is at } \mu_{i,k}$ ;
8    $\delta I_j = w_j \sum_{j=1}^n \Delta I_{ij}$  ;
9  $\pi^* = \text{Pick policy from } \Pi \text{ that has maximum gain } \delta I$  ;
10 return  $\pi^*$ ;
```

---

---

**Algorithm 6:** Calculating Expected Information Gain

---

```
1 Input:  $\pi_j, m_{i,k}$ 
2 Output:  $\Delta I_{ij}$ 
3  $x_0 = \mu_{i,k}$  ;
4  $n_0 = \text{Current number of belief modes}$ ;
5  $\Delta I_{ij} = 0$  ;
6 forall the  $u_k \in \pi_i$  do
7    $x_{k+1} = f(x_k, u_k, 0)$  ;
8    $z_{k+1} = h(x_{k+1}, 0)$  ;
9   Propagate individual Kalman filters with control  $u_k$ 
   and most likely observation  $z_{k+1}$  ;
10  if Any belief mode is in collision configuration then
11     $\Delta I_{ij} = \Delta I_{ij} - \frac{c_{fail}}{k}$ ;
12    break;
13  $n_T = \text{Number of belief modes after simulation}$ ;
14  $\Delta I_{ij} = \Delta I_{ij} + n_T - n_0$  ;
15 return  $\Delta I_{ij}$ ;
```

---

*Assumption 2:* The map does not change during the execution of the planner.

*Proposition 1:* Under Assumptions 1 and 2, given any initial multi-modal belief  $b_0 = \sum_i w_{i,0} m_{i,0}$ , the receding horizon planner M3P drives the belief process into a unimodal belief  $b_T = m_T \approx \mathcal{N}(\mu_T, \Sigma_T)$  in some finite time  $T$ .

*Proof:* Suppose that the robot is at the initial belief  $b_0$ . Suppose we choose the plan  $\mu_{i^*}$  that results in the most information gain as required by the M3P planner. The plan  $\mu_{i^*}$  can be applied to all the modes at least for some finite period of time in the future, since if it cannot be applied, then we immediately know that the robot is not at mode  $i^*$  and thus, there is a disambiguation whereby mode  $i^*$  is discarded. Once the plan  $\mu_{i^*}$  is executed, there are only 2 possibilities: 1) The robot is able to execute the entire plan  $\mu_{i^*}$  till the end, or 2) the plan becomes infeasible at some point of its execution.

In case 1, due to Assumption 1, we will know for sure that the

robot was at mode  $i^*$  and the belief collapses into a unimodal belief thereby proving the result. In case 2, due to Assumption 2, we know that the robot could not have started at mode  $i^*$  and thus, the number of modes is reduced by at least one. After this disambiguation, we restart the process as before and we are assured that atleast one of the modes is going to be disambiguated and so on. Thus, it follows given that we had a finite number of modes to start with, the belief eventually converges to a unimodal belief. Further, since each of the disambiguation epochs takes finite time, a finite number of such epochs also takes a finite time, thereby proving the result. ■

The above result shows that the basic M3P algorithm will stabilize the belief process to a unimodal belief under Assumptions 1 and 2. In order to show that a particular instantiation of the planner, such as the target node based planner NBM3P detailed above, stabilizes the belief to a single mode, we need to show that the particular planner satisfies Assumption 1. It can be seen that under NBM3P, by design, the robot reaches a unique disambiguating location if it started in a particular mode, and thus, Assumption 1 is satisfied. Hence, this leads to the following corollary.

*Corollary 1:* The Node based receding horizon planner, NBM3P, drives any initial multi-modal belief into a unimodal belief in finite time, under Assumption 2.

*Remark 1:* The above results hold if we choose the disambiguating policy at random, and not as the optimal one in the sense of the number of modes disambiguated. However, in practice we see that the optimal disambiguating policy gives us better performance.

#### IV. SIMULATION RESULTS

We present simulation results for a 2D robot. The simulations represent a motion planning scenario wherein the robot is tasked to go from a start to a goal location in an environment where there is symmetry. We initially rely on FIRM [3] to provide a feedback policy. However, en-route to the goal, the robot is kidnapped to an unknown location and it cannot localize. Thus, it relies on the non-Gaussian planner NBM3P described previously to localize. Once the robot is localized, its new belief is connected to the existing FIRM graph and we find a new feedback policy to complete the task. We abstract the geometry/appearance of the environment by using passive visual beacons that have IDs associated to them. We place multiple landmarks with the same IDs in different locations, thus creating the ‘illusion’ of multiple-data associations. All simulations were carried out on a Macbook Pro laptop with an Intel Core-i5 2.6GHz CPU and 8GB of RAM running Ubuntu 14.04, we use the Open Motion Planning Library [15] as the software back-end.

*A supplementary video is provided [1] that clearly depicts every stage of the simulations with a robot operating in a virtual environment. The key steps in the simulation are:*

- 1) Generate an initial policy from start to goal.

- 2) Keep tracking innovation, large change causes robot to detect lost state.
- 3) Switch from Gaussian to a GMM representation of the underlying belief.
- 4) Generate information gathering actions using M3P that are able to localize the state.
- 5) Switch back to uni-modal Gaussian and connect the belief to existing FIRM roadmap.
- 6) Find new policy to goal.

#### A. Motion Model

We simulate a 2D ground robot, the kinematics of which are represented by a unicycle.

$$x_{k+1} = f(x_k, u_k, w_k) = \begin{pmatrix} x_k + (V_k + n_v)\delta t \cos \theta_k \\ y_k + (V_k + n_v)\delta t \sin \theta_k \\ \theta_k + (\omega_k + n_\omega)\delta t \end{pmatrix}, \quad (5)$$

where  $x_k = (x_k, y_k, \theta_k)^T$  describes the robot state (position and yaw angle).  $u_k = (V_k, \omega_k)^T$  is the control vector consisting of linear velocity  $V_k$  and angular velocity  $\omega_k$ . We denote the process noise vector by  $w_k = (n_v, n_\omega)^T \sim \mathcal{N}(0, \mathbf{Q}_k)$ .

#### B. Observation Model

Our observation model is based on passive beacons/landmarks which can be pinged to measure their relative range and an associated ID tag. This is parallel to a physical implementation using a monocular camera and passive Augmented Reality (AR) beacons that can be detected by the camera. We use the visual AR model to simplify the data association problem.

Let the location of the  $i$ -th landmark be denoted by  ${}^i\mathbf{L}$ . The displacement vector  ${}^i\mathbf{d}$  from the robot to  ${}^i\mathbf{L}$  is given by  ${}^i\mathbf{d} = [{}^i d_x, {}^i d_y]^T := {}^i\mathbf{L} - \mathbf{p}$ , where  $\mathbf{p} = [x, y]^T$  is the position of the robot. Therefore, the observation  ${}^i z$  of the  $i$ -th landmark can be modeled as,

$${}^i z = {}^i h(x, {}^i v) = [\|{}^i\mathbf{d}\|, \text{atan2}({}^i d_y, {}^i d_x) - \theta]^T + {}^i v, \quad (6)$$

The observation noise is zero-mean Gaussian such that  ${}^i v \sim \mathcal{N}(\mathbf{0}, {}^i\mathbf{R})$  where  ${}^i\mathbf{R} = \text{diag}((\eta_r \|{}^i\mathbf{d}\| + \sigma_b^r)^2, (\eta_\theta \|{}^i\mathbf{d}\| + \sigma_b^\theta)^2)$ . The quality of sensor reading decreases as the robot gets farther from the landmarks. The parameters  $\eta_r$  and  $\eta_\theta$  determine this dependency, and  $\sigma_b^r$  and  $\sigma_b^\theta$  are the bias standard deviations. The robot observes only those landmarks that fall within its sensor range  $r_{\text{sensor}}$ .

#### C. Scenario

The environment as shown in Fig. 3 represents a warehouse floor where 6 corridors C1-6 appear exactly alike. These corridors open into two passages which also appear alike. There are unique landmarks at location L1. The robot is put at the start location S and tasked to reach goal location G as shown. Using FIRM, we generate the feedback policy from S to G. The blue ellipse and arrow mark the region where

the robot will be kidnapped and where it will be displaced to respectively First, we generate an initial feedback policy from S to G. When the robot is en-route to G (Fig. 4a) it is kidnapped and placed in corridor C2 as shown in Fig. 4b. A new multi-modal belief is generated which took 51.23s to compute.

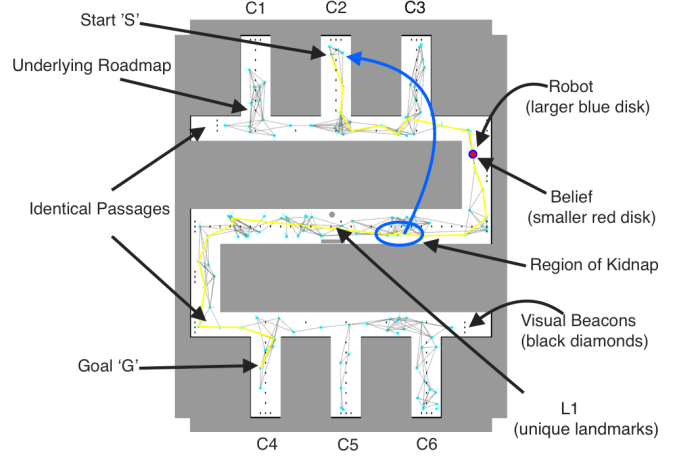


Fig. 3: The environment depicting obstacles (gray) and free space (white). The cyan disks represent the nodes in the FIRM roadmap. The robot is en-route to the goal before kidnapping and the initial plan under FIRM is demarcated by yellow. The blue ellipse marks the region where the robot will be kidnapped (not known to robot) and the final location after kidnapping (in C2) is marked by the blue arrow (begins inside the ellipse).

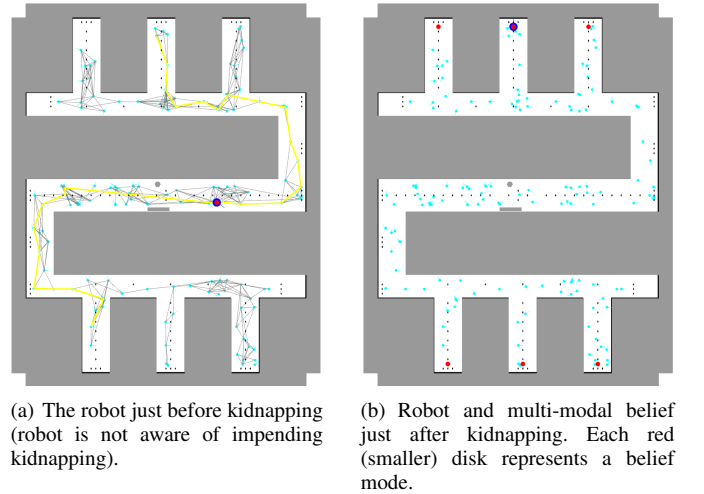


Fig. 4: Before and after kidnapping.

We analyze the recovery from kidnapping at four different time steps:

- 1)  $t_1$ : Fig. 5a shows the initial candidate policies after the new belief is sampled. The policy was computed in 58.76s.



- 2)  $t_2$ : Fig. 5a shows that on exiting the corridors, the modes  $m_3$  and  $m_4$  that originated in  $C3$  and  $C4$  expect to see a corner to their left and are rejected as this is not observed by the robot. A new policy is then computed which took 12.48s.
- 3)  $t_3$ : In Fig. 5c we see the new candidate policy. This eventually leads the robot to move forward which results in the modes  $m_1$  and  $m_6$  that exit  $C1$  and  $C6$  to be rejected as the robot expects to see a corner. Once these two modes are rejected a new policy is computed which took 3.7s.
- 4)  $t_4$ : Fig. 5d shows the final two remaining modes approaching the unique landmark which eventually leads to a uni-modal belief localized to the true state.

Fig. 6a shows the belief localized, once this belief is added to the existing FIRM roadmap, a new policy guides the robot to the goal (Fig. 6b).

#### D. Discussion

It is seen that a highly intuitive behavior emerges which guides the robot to seek disambiguating information such that it can sequentially reject the incorrect hypothesis about its state. The open-loop control actions are regenerated every time a Gaussian mode is rejected or a constraint violation is foreseen. The time to re-plan reduces drastically as the number of modes reduce ( $O(n^2)$ ). Thus, the first few actions are the hardest which is to be expected as we start off with a large number of hypotheses. Finally, the planner is able to localize the robot safely.

#### V. CONCLUSION

In this work, we studied the problem of motion planning for a mobile robot when the underlying belief state is non-Gaussian in nature. A non-Gaussian belief can result from uncertain data associations or due to unknown initial conditions among others. Our main contribution in this work is a planner M3P that generates a sequentially disambiguating policy, which leads the belief to converge to a uni-modal Gaussian. We are able to show in simulation that the robot is able to recover from a kidnapped state and execute its task in environments that present multiple uncertain data associations. Our next step would be to demonstrate the same on a physical system and prove its robustness in real-life situations.

#### BIBLIOGRAPHY

- [1] Video of M3P plan execution in simulation. <https://www.youtube.com/watch?v=8kldm34Ya4I>.
- [2] Aliakbar Agha-mohammadi, Saurav Agarwal, Aditya Mahadevan, Suman Chakravorty, Daniel Tomkins, Jory Denny, and Nancy Amato. Robust online belief space planning in changing environments: Application to physical mobile robots. In *IEEE Int. Conf. Robot. Autom. (ICRA)*, Hong Kong, China, 2014.
- [3] Aliakbar Agha-mohammadi, Suman Chakravorty, and Nancy Amato. FIRM: Sampling-based feedback motion

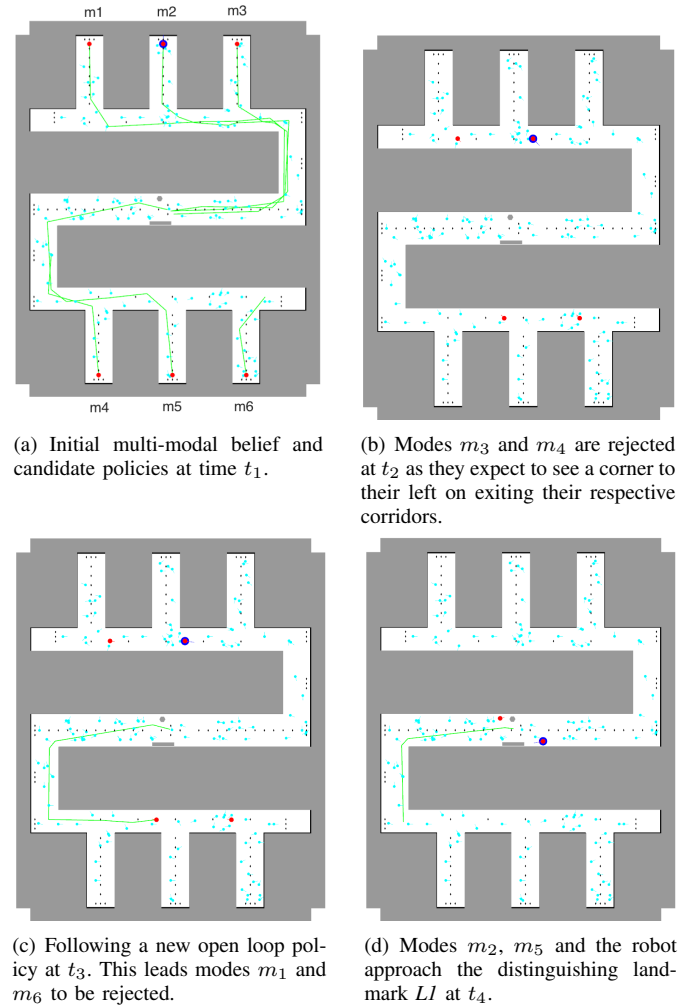


Fig. 5: Candidate policies and belief modes during recovery from kidnapping.

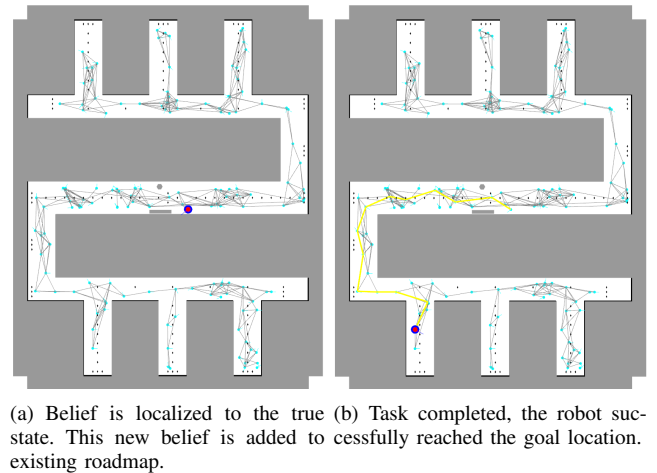


Fig. 6: Recovery from kidnapping

planning under motion uncertainty and imperfect measurements. *International Journal of Robotics Research*



- (*IJRR*), 33(2):268–304, 2014.
- [4] Adam Bry and Nicholas Roy. Rapidly-exploring random belief trees for motion planning under uncertainty. In *ICRA*, pages 723–730, 2011.
  - [5] Pratik Chaudhari, Sertac Karaman, David Hsu, and Emilio Frazzoli. Sampling-based algorithms for continuous-time POMDPs. In *the American Control Conference (ACC)*, Washington DC, 2013.
  - [6] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *International Journal of Robotics Research*, 30(7):846–894, June 2011.
  - [7] L.E. Kavraki, P. Svestka, J.C. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.
  - [8] H. Kurniawati, T. Bandyopadhyay, and N.M. Patrikalakis. Global motion planning under uncertain motion, sensing, and environment map. *Autonomous Robots*, 33(3):255–272, 2012.
  - [9] Cody Kwok, Dieter Fox, and Marina Meila. Adaptive real-time particle filters for robot localization. In *Robotics and Automation, 2003. Proceedings. ICRA’03. IEEE International Conference on*, volume 2, pages 2836–2841. IEEE, 2003.
  - [10] Steve Lavalley and J. Kuffner. Randomized kinodynamic planning. *International Journal of Robotics Research*, 20(378-400), 2001.
  - [11] R. Platt. Convex receding horizon control in non-Gaussian belief space. In *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2012.
  - [12] R. Platt, L. Kaelbling, T. Lozano-Perez, , and R. Tedrake. Efficient planning in non-Gaussian belief spaces and its application to robot grasping. In *Proc. of International Symposium of Robotics Research, (ISRR)*, 2011.
  - [13] Robert Platt, Leslie Kaelbling, Tomas Lozano-Perez, and Russ Tedrake. Non-gaussian belief space planning: Correctness and complexity. In *ICRA*, 2012.
  - [14] Sam Prentice and Nicholas Roy. The belief roadmap: Efficient planning in belief space by factoring the covariance. *International Journal of Robotics Research*, 28(11-12), October 2009.
  - [15] Ioan A. Şucan, Mark Moll, and Lydia E. Kavraki. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, December 2012. doi: 10.1109/MRA.2012.2205651. <http://ompl.kavrakilab.org>.
  - [16] Jur van den Berg, Pieter Abbeel, and Ken Goldberg. LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information. In *Proceedings of Robotics: Science and Systems (RSS)*, June 2010.